Access vs. Bandwidth in Codes for Storage

Itzhak Tamo*,†, Zhiying Wang* and Jehoshua Bruck*

*Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125, USA
†Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva 84105, Israel

{tamo, zhiying, bruck}@caltech.edu

Abstract—Maximum distance separable (MDS) codes are widely used in storage systems to protect against disk (node) failures. An \( (n, k, l) \) MDS code uses \( n \) nodes of capacity \( l \) to store \( k \) information nodes. The MDS property guarantees the resiliency to any \( n-k \) node failures. An optimal bandwidth (resp. optimal access) MDS code communicates (resp. accesses) the minimum amount of data during the recovery process of a single failed node. It was shown that this amount equals a fraction of \( 1/(n-k) \) of data stored in each node. In previous optimal bandwidth constructions, \( l \) scaled polynomially with \( k \) in codes with asymptotic rate \( < 1 \). Moreover, in constructions with a constant number of parity, i.e. rate approaches \( 1 \), \( l \) is scaled exponentially w.r.t. \( k \). In this paper, we focus on the practical case of \( n-k = 2 \), and ask the following question: Given the capacity of a node \( l \) what is the largest (w.r.t. \( k \) ) optimal bandwidth (resp. access) \((k + 2, k, l) \) MDS code. We give an upper bound for the general case, and two tight bounds in the special cases of two important families of codes.

I. INTRODUCTION

Erasure-correcting codes are the basis for widely used storage systems, where disks (nodes) correspond to symbols in the code. An important family of codes is the Maximum distance separable (MDS) codes, which provide an optimal resiliency to erasures for a given amount of redundancy. Namely, an MDS code with \( r \) redundancy (parity) symbols can recover the information from any \( r \) symbol erasures. Because of this storage efficiency, MDS codes are highly favorable, and a lot of research has been done to construct them. Examples of MDS codes are the well-known Reed Solomon codes and EVENODD. It is evident that in the case of \( r \) erasures, one needs to communicate all the surviving information during the recovery process. However, although the MDS codes used in practice are resilient to more than a single erasure, i.e. number of parity nodes \( r > 1 \), the practical and more interesting question is; what is the minimum repair bandwidth in a single node erasure. The repair bandwidth is defined as the amount of information communicated during the recovery process. This question has received much interest recently due to both its practical and theoretical importance. From a practical viewpoint, decreasing the repair bandwidth shortens both the recovery process and the inaccessibility time of the erased information. Moreover, from a theoretical perspective, this question has deep connections to the widely used interference alignment technique and network coding.

A. The Problem

The problem of efficient recovery was defined by Dimakis et al in [4]. It considers a file of size \( M \) divided into \( k \) equally sized chunks stored using an \( (n, k, l) \) MDS code, where \( n \) is the number of nodes, each of capacity \( l = \frac{M}{n} \). The first \( k \) nodes, which are referred to as the systematic nodes, store the raw information. The later \( r = n-k \) nodes are the parity nodes which store a function of the raw information. Since the code is MDS, it can tolerate any loss of up to \( r \) nodes. However, the more common scenario is the failure (erasure) of only one node. [4] proved that

\[
l \geq \frac{n-1}{n-k},
\]

is a lower bound on the repair bandwidth for an \( (n, k, l) \) MDS code. In particular, for a code with \( r = 2 \) parities, the repair bandwidth for each surviving node is \( \frac{1}{2} \), i.e. at least one half of its information needs to be communicated. Note that recovery is possible since the code is resilient to more than one erasure, and a repair strategy of communicating the entire remaining information suffices. An MDS code is termed optimal bandwidth if it achieves the lower bound in (1) during the recovery process of all of its systematic nodes. Figure 1 shows an optimal bandwidth \((6,4,2)\) MDS code. For recovering an erased node, one bit of information is transmitted to the repair center from each surviving node. In some applications such as data centers, reading (accessing) the information is more costly than transmitting it. Therefore during a recovery process, the need to transmit data that is a function of a large portion of the information stored within a node, can cause a bottleneck. For example, node \( N1 \) needs to access its entire stored information, for it to calculate \( a + w \), during the recovery process of node \( N3 \). Therefore, in a large scale storage systems, one might need to minimize not only the amount of information transmitted but also the number of accessed information elements. An optimal access MDS

\[ N1 \begin{array}{llll} \begin{array}{llll} a & b & c & d \\
N2 & w & x & y & z \end{array} & a+b+c+d \end{array} N3 & \begin{array}{llll} \begin{array}{llll} \end{array} \end{array} N4 & \begin{array}{llll} \begin{array}{llll} \end{array} \end{array} Parity 1 & \begin{array}{llll} \begin{array}{llll} \end{array} \end{array} Parity 2 & \begin{array}{llll} \begin{array}{llll} \end{array} \end{array} a+5w+b+2c+5d \end{array} \begin{array}{llll} \begin{array}{llll} 3w+2b+3x+4y+5z \end{array} \end{array} \end{array} Figure 1. \((6,4,2)\) MDS code with optimal bandwidth over the field \( \mathbb{F}_2 \). Nodes \( N1, N2, N3, N4 \) are systematic and the last 2 nodes are parity nodes. For recovering node \( N1 \), (resp. \( N2 \)) transmit the first (second) row from each surviving node. For recovering node \( N3 \) transmit from each surviving node the sum of its two elements. For recovering node \( N4 \) transmit the sum of the first row and twice the second row from Parity 2, and the sum of the first row and four times the second row from the rest.

1The relaxed requirement of optimal recovery only for the systematic nodes is reasonable, because the number of parity nodes in most storage systems is negligible compared to systematic nodes. Moreover, in an erasure of a systematic node, the raw information is not accessible as opposed to a parity node erasure.

---

\( N2 \) \( a+5w+b+2c+5d \)

\( a+b+c+d \)

\( b \)

\( z \)

\( N3 \)

\( x \)

\( N4 \)

\( y \)

\( w+x+y+z \)
code is an optimal bandwidth code that transmits only the elements it accesses. By definition, any optimal access code is also an optimal bandwidth code. The shortened code restricted to nodes \( \{N_1, N_2, \text{Parity 1}, \text{Parity 2} \} \) in Figure 1 is an example of an optimal access \((4, 2, 2)\) MDS code.

In a value’s update of a stored element, one needs to update each parity node at least once. To avoid an overload on the system during a frequent operation such as updating, one needs to design an optimal update code, that updates exactly once in each parity node, when an element changes its value. For example in Figure 1 the shortened code restricted to nodes \( \{N_3, N_4, \text{Parity 1}, \text{Parity 2} \} \) is an optimal update and optimal bandwidth \((4, 2, 2)\) MDS code, because updating any of the elements \(c, d, y, z\) will require updating exactly one element in each of the parity nodes.

Various codes [4], [7]–[9], [11], [15]–[17] were constructed with the goal of achieving optimal bandwidth, however these constructions all have low rate, i.e., \(k/n \leq 1/2\). In [9], [11], [17] the key idea was using vector coding. Namely, each symbol in a codeword is a vector and not scalar as in “standard” codes. Specifically [9], [11] constructed optimal bandwidth \((2k, k, k)\) MDS codes. Using interference alignment, it was shown in [3] that the bound in (1) is asymptotically achievable also for high rate codes \((k/n \geq 1/2)\). The question of existence of optimal bandwidth codes with high rate was resolved in several constructions [1], [2], [5], [6], [12], [13], [18]. The constructions have an arbitrary number of parity nodes \(r\), however when \(r\) is constant, i.e. rate approaching 1 in all of the constructions \(k = O(\log, l)\), i.e., the capacity \(l\) scales exponentially with the number of systematic nodes \(k\).

<table>
<thead>
<tr>
<th>Optimal Bandwidth</th>
<th>Optimal Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k = 1 + \log l, \sqrt{\ast}) ([12])</td>
<td>(k = 1 + \log l, \sqrt{\ast}) ([12])</td>
</tr>
<tr>
<td>(3 \log l \leq k \leq h_{l/2} + 1, \ast)</td>
<td>(k = 2 \log l, \sqrt{\ast}) ([1])</td>
</tr>
</tbody>
</table>

TABLE I
Summary of known results on maximum number of information nodes \(k\) in an \((k + 2, k, l)\) MDS code. \(\sqrt{\ast}\) indicates a tight bound, \(\ast\) indicates a new upper bound. The references refer to previously known lower bounds.

B. Our Contribution
Our main goal in this paper is to understand the relation between \(l\) the capacity of each node, and the number of systematic nodes \(k\). More precisely, given the capacity of the node \(l\), what is the largest integer \(k\), such that there exists an optimal bandwidth or optimal access \((k + 2, k, l)\) MDS code. We will derive three upper bounds on \(k\) as a function of only \(l\), for different families of codes. To derive the bounds, we use 3 different combinatorial techniques. The first bound considers the general problem, where no requirements on the MDS code are imposed except the optimal bandwidth property. The bound is derived by defining an appropriate set of multivariate polynomials. We proceed by deriving a tight bound for optimal bandwidth MDS codes with diagonal encoding matrices. These codes are a part of an important family of codes with an optimal update property. The last result provides a tight bound on optimal access MDS codes. Table I summarizes the known results together with our new results. Due to space limitation, we only consider the more practical case of \(r = 2\) parities, although the results can be extended to an arbitrary \(r\). Some of the proofs are also omitted and can be found in [14].

C. Settings and Notation
Consider a file of size \(M = kl\), divided into \(k\) nodes of capacity \(l\). Each systematic node \(1 \leq i \leq k\) is represented by an \(l \times 1\) vector \(a_i \in F_p^l\) over the field \(F_p\). We construct an \((k + 2, k, l)\) MDS code by adding parity nodes \(k + 1\) and \(k + 2\), which will give the resiliency to node erasures. Parity node \(k + i\) for \(i \in \{1, 2\}\) stores the information vector \(a_{k+i}\) of length \(l\) over \(F_p\), and is defined as

\[
a_{k+1} = \sum_{i=1}^{k} A_i a_i, \text{ and } a_{k+2} = \sum_{i=1}^{k} B_i a_i.
\]

Where \(A_i, B_i\) are square matrices of order \(l\), which are called the encoding matrices. Note that the code has a systematic structure, i.e., the first \(k\) nodes store the information itself, and not a function of it. Therefore, the code is defined uniquely by the matrix

\[
\begin{bmatrix}
A_1 & A_2 & \ldots & A_k \\
B_1 & B_2 & \ldots & B_k
\end{bmatrix}.
\]

(2)

The code is called an MDS if it can recover from any \(2\) node erasures, which is equivalent to any \(1 \times 1\) and \(2 \times 2\) block sub matrix in (2) is invertible. In a scenario of an erasure of a systematic node \(i, 1 \leq i \leq k\), a linear combination of the information stored in the parity nodes is transmitted in order to recover the lost data. Namely, the parity nodes \(k + 1\), and \(k + 2\), project their data on the recovering subspaces \(S_i, T_i\) of dimension \(l/2\), respectively. In other words, the transmitted information from parity nodes \(k + 1, k + 2\) during the recovery process of systematic node \(i \in [k]\) is the projection \(S_i a_{k+1}, T_i a_{k+2}\) respectively, where each subspace \(S_i, T_i\) is represented by an \(l/2 \times l\) matrix whose rows form a basis of the subspace. In general, each subspace will be represented by a matrix whose rows form a basis of the subspace. Recovering the lost information is possible if

\[
\text{rank } \begin{bmatrix} S_i A_j \\ T_i B_j \end{bmatrix} = l.
\]

(3)

Moreover, the recovery process is optimal bandwidth if for each \(j \neq i\),

\[
\text{rank } \begin{bmatrix} S_i A_j \\ T_i B_j \end{bmatrix} = \frac{l}{2}.
\]

(4)

Similar conditions were derived in [9]. Therefore an optimal bandwidth algorithm for the systematic nodes is defined by the pairs of recovering subspaces \((S_i, T_i)\) that satisfy (3), (4) for \(1 \leq i \leq k\). For any integer \(k\) denote by \([k] = \{1, \ldots, k\}\). For simplicity, we will assume that the capacity of each node \(l\), is a power of 2, and all the logarithms are of base 2.
The remainder of the paper is organized as follows. Section II defines the subspace property for a set of matrices, and shows the equivalency of this property to optimal bandwidth codes. Section III provides an upper bound for the most general case, i.e., an MDS code with optimal bandwidth property. We proceed in Section IV where a tight bound is derived for codes with diagonal encoding matrices. In Section V a tight bound for codes with optimal access property is derived. We conclude with a summary in Section VI.

II. THE SUBSPACE PROPERTY

In this section we define the Subspace Property for a set of invertible matrices, and show its connection to optimal bandwidth MDS codes.

**Subspace Property:** A set of invertible matrices \( C_1, ..., C_k \) of order \( l \), is said to satisfy the subspace property, if there exists a set of subspaces \( S_1, ..., S_k \) of dimension \( \frac{l}{2} \), such that for any \( 1 \leq i, j \leq k \),

\[
S_i C_j \cap S_i = \begin{cases} \{0\} & i = j, \\ S_i & \text{else,} \end{cases}
\]

where \( \{0\} \) is the zero subspace, and \( S_i C_j \) is image of the action of the invertible matrix \( C_j \) on the \( l/2 \)-dimensional subspace \( S_i \). We start with a theorem that shows we can assume w.l.o.g. that an optimal bandwidth MDS code has a certain structure where a subset of the encoding matrices satisfies the subspace property.

**Theorem 1** There exists an optimal bandwidth \((k + 2, k, l)\) MDS code if and only if there exists an optimal bandwidth MDS code with the same parameters and encoding matrices

\[
\left[ \begin{array}{ccc} I & \ldots & I \\ C_1 & \ldots & C_{k-1} \\ & \ldots & \end{array} \right],
\]

such that the matrices \( C_1, ..., C_{k-1} \), satisfy the subspace property. Moreover, from any set \( C_1, ..., C_{k-1} \) of invertible matrices of order \( l \) that satisfy the subspace property, we can construct over a field large enough, an optimal bandwidth, \((m + 2, m, l)\) MDS code, where \( k - 1 \leq m \leq k \).

**Proof:** Proof is omitted.

By the previous theorem, we assume that any optimal bandwidth MDS code is of the form (6), with a set of matrices \( \{C_i\} \) that satisfy the subspace properties. Therefore, we recast the recovering problem into a problem on a set of matrices which satisfy this property.

From the last Theorem we have the following Corollary.

**Corollary 2** Let \( k \) be the largest integer such that there exists an optimal bandwidth, \((k + 2, k, l)\) MDS code. Let \( s \) be the size of the largest set of invertible matrices of order \( l \) that satisfy the subspace property, then \( s \leq k \leq s + 1 \).

**Proof:** Proof is omitted.

Theorem 1 shows that constructing an optimal bandwidth MDS code with 2 parities is equivalent to finding a set of invertible matrices that satisfy the subspace property.

III. UPPER BOUND ON THE NUMBER OF NODES IN AN OPTIMAL BANDWIDTH MDS CODE

We start with the most general problem which seems to be the most difficult of them all. We impose no constraints on the encoding matrices and the recovering subspaces. We derive an upper bound on the number of information nodes \( k \) in an optimal bandwidth \((k + 2, k, l)\) MDS code. The bound is a function of only the capacity of the node \( l \), regardless of the field size being used.

Before we prove the upper bound, for a set of indexes of equal size \( l, J \) define \( \text{det}(B)_{l,J} \) to be the determinant of the matrix of \( B \) restricted to rows \( l \) and columns \( J \).

**Theorem 3** Let \( C_1, ..., C_k \) be a set of matrices of order \( l \) over a field \( \mathbb{F} \), together with a set of subspaces \( S_1, ..., S_k \) of dimension \( \frac{l}{2} \) each, such that the subspace property is satisfied, then

\[ k \leq l \left( \frac{l}{1/2} \right). \]

**Proof:** Assume that each subspace \( i \) is represented by a matrix \( S_i \), of dimension \( \frac{l}{2} \times l \). For any \( 1 \leq i \leq k \) the matrix

\[
\left( \begin{array}{c} S_i \\ S_i C_i \end{array} \right)
\]

is of full rank, hence there exists a set of indexes \( I \subset [l] \) of size \( \frac{l}{2} + 1 \) such that the \( \frac{l}{2} + 1 \times \frac{l}{2} + 1 \) sub matrix restricted to the set of rows and columns, \( \left[ \frac{l}{2} + 1 \right] \) and \( I \) respectively, is invertible. Namely,

\[
\text{det} \left( \begin{array}{c} S_i \\ S_i C_i \end{array} \right)_{[\frac{l}{2} + 1],I} \neq 0.
\]

Moreover, since for any \( j \neq i \),

\[
\text{rank} \left( \begin{array}{c} S_i \\ S_i C_i \end{array} \right) = \frac{l}{2}.
\]

The sub matrix restricted to the same set of rows and columns is not of full rank (note that for distinct \( i \)'s the set of indexes \( I \) might be different). Hence, the polynomial \( f_j : \mathbb{F}^{\frac{l}{2} \times l} \rightarrow \mathbb{F} \), defined by

\[
f_j(S) = \text{det} \left( S S C_i \right)_{[\frac{l}{2} + 1],I},
\]

satisfies,

\[
f_i(S_j) = \begin{cases} \neq 0 & i = j, \\ 0 & \text{otherwise}. \end{cases}
\]

Define two sets of polynomials

\[
T_1 = \left\{ \text{det} \left( \begin{array}{ccc} x_{1,1} & \cdots & x_{1,l} \\ \vdots & \ddots & \vdots \\ x_{l,1} & \cdots & x_{l,l} \end{array} \right) : J \in \left( \left[ \frac{l}{2} \right] \right) \right\},
\]

and \( T_2 = \{ x_{1,i} : 1 \leq i \leq l \} \), where \( \left( \frac{l}{2} \right) \) is the set of \( l/2 \)-subsets of \( [l] \). Using (7) check that the polynomials \( \{f_i\} \) are linearly independent, and they are spanned by the set

\[
T_1 \cdot T_2 = \{ h \cdot g : h \in T_1, g \in T_2 \},
\]
which is of size at most $|T_1| \cdot |T_2| = l^{(1/2)}$. Therefore, $k = \lfloor \log l \rfloor$.

**Corollary 4** Let $k$ be the largest integer, such that there exists an optimal bandwidth $(k + 2, k, l)$ MDS code, then

$$3 \log l \leq k \leq 1 + l\left(\frac{1}{\log l}\right).$$

**Proof:** The upper bound is a consequence of Corollary 2 and Theorem 3, the lower bound is given by the code constructed in [19].

As one can notice, there exists a big gap between the upper and the lower bound. We conjecture that the lower bound is more accurate, and in fact $k = \theta(\log l)$.

We proceed by giving a tight bound for the number of systematic nodes $k$ in the case where all the encoding matrices are diagonal.

**IV. UPPER BOUND FOR DIAGONAL ENCODING MATRICES**

One of the most common operations in the maintenance of storage systems is updating. Namely, a certain element has changed its value, and that needs to be updated in the system. Since the code is an MDS, each parity node is a function of the entire systematic nodes. Therefore, in a single update, each parity node needs to be updated at least in one of the elements it stores. An optimal update code is one that needs to update each parity node exactly once in an update of any information element. We derive a tight bound in a special case of an optimal update code, where all the encoding matrices are diagonal. We begin with a simple lemma on an entropy.

**Lemma 5** Let $X$ be a random variable such that for any possible outcome $x$, $P(X = x) \leq \frac{1}{2}$, then its entropy satisfies $H(X) \geq 1$.

**Proof:** Proof is omitted.

Before we proceed to prove the upper bound, recall that the meet of two partitions $X, Y$ of some set, is defined as,

$$X \cap Y = \{x \cap y : x \in X, y \in Y\}.$$  

Moreover, for a set of indices $x$ denote by $\text{span}(e_x) = \text{span}(e_i : i \in x)$, where $e_i$ is the $i$-th vector in the standard basis.

**Theorem 6** Let $\{C_i\}_{i=1}^k$ be a set of invertible and simultaneously diagonalizable matrices of order $l$, that satisfy the subspace property, then $k \leq \log l$.

**Proof:** Since the subspace property is preserved under similarity transformation, we can assume that all the matrices $C_1, ..., C_k$ are diagonal and the standard basis $\{e_1, ..., e_l\}$ is a set of eigenvectors for all the matrices. Each matrix $C_i$ defines a partition $X_i$ of $[l]$, by $m, n \in [l]$ are in the same set of the partition, iff the corresponding vectors $e_m$ and $e_n$ have the same eigenvalue in $C_i$. Let $j \in [k]$, set $S = S_j$ and denote the meet of the partitions

$$X = \bigcap_{j \neq j} X_j.$$  

It is clear that a vector $v \neq 0$ is an eigenvector for all the matrices $C_i, i \neq j$ iff $v \in \text{span}(e_x)$, for some set $x$ in the partition $X$. Assume $S$ is represented in its reduced row echelon form. Since $S$ is an invariant subspace for $C_i, i \neq j$, it is clear that each row vector in $S$ is an eigenvector for each of the matrices $C_i, i \neq j$. Hence, for each row of $S$, the set of indices of the nonzero entries is contained in some set of the partition $X$. Therefore,

$$S = \oplus_{x \in X} S_x,$$

where $S_x = S \cap \text{span}(e_x)$. Let $x \in X$, and note that $\text{dim}(S_x) = \frac{|x|}{2}$. Since, if $S_x > \frac{|x|}{2}$ and the fact that $\text{span}(e_x)$ is also an invariant subspace of $C_j$ we get $S_xC_jS_x \subset \text{span}(e_x)$. Hence $S_xC_j \cap S_x \neq \{0\}$, which contradicts the subspace property. Moreover, since

$$\frac{l}{2} = \text{dim}(S) = \sum_{x \in X} \text{dim}(S_x) \leq \sum_{x \in X} \frac{|x|}{2} = \frac{l}{2},$$

we get that $\text{dim}(S_x) = \frac{|x|}{2}$. Denote the partition of $x$ by $X_j$ as $x = \{x \cap y : y \in X_j\}$. We claim that

$$\max_{x \in X} |z| \leq \frac{|x|}{2}. \quad (8)$$

Assume to the contrary, that $z \in x, |z| > \frac{|x|}{2}$. Note that each $v \in \text{span}(e_x)$ is an eigenvector also for $C_j$, and

- $S_x, \text{span}(e_x) \subseteq \text{span}(e_x)$,
- $\text{dim}(S_x) = \frac{|x|}{2}, \text{dim}(\text{span}(e_x)) > \frac{|x|}{2}$

Hence, $S_x \cap \text{span}(e_x) \neq 0$, i.e., $S_x$ contains an eigenvector of $C_j$ which contradicts the subspace property, because

$$SC_j \cap S \supseteq (S_x \cap \text{span}(e_x))C_j \cap (S_x \cap \text{span}(e_x)) = (S_x \cap \text{span}(e_x)) \cap (S_x \cap \text{span}(e_x)) = (S_x \cap \text{span}(e_x)) \neq \{0\}.$$  

Pick randomly and with equal probability a vector $v$ from the standard basis, and define for $i \in [k]$ the random variable $Y_i$ to be the eigenvalue corresponding to the eigenvector $v$ in $C_i$. From (8) and by Lemma 5 we conclude that the entropy of $Y_i$ satisfies

$$H(Y_i|Y_i, l \neq j) \geq 1. \quad (9)$$

Therefore,

$$\log l = H(v) = H(v, Y_1, ..., Y_k) \geq H(Y_1, ..., Y_k) + H(v|Y_1, ..., Y_k) \geq \sum_{j=1}^{k} H(Y_1, ..., Y_k) \geq \sum_{j=1}^{k} 1 = k,$$

where the last inequality follows from (9).

**Corollary 7** Let $k$ be the largest integer such that there exists an optimal bandwidth $(k + 2, k, l)$ MDS code with diagonal encoding matrices, then $k = 1 + \log l$.

**Proof:** The lower bound is given by the code constructed in [12].
Note that when restricting to diagonal encoding matrices, there is no difference if the code is an optimal access or optimal bandwidth (see Table I). In the next section, we show that these two properties are not equivalent in the general case.

V. UPPER BOUND ON THE NUMBER OF NODES FOR OPTIMAL ACCESS

Recall that in an optimal bandwidth MDS codes the transmitted information can be a function of the entire information in the node. Namely, in order to generate the transmitted data, one has to access all the information stored in the node which, of course, can be an expensive task. An optimal access code is an optimal bandwidth code that transmits only the elements it accesses. The property of Optimal Access is equivalent to that each recovering subspace $S_i$ is spanned by an $1/2$-subset of the standard basis $e_1, ..., e_l$, i.e., $S_i = \text{span}(e_m : m \in I)$ for some $I$ an $1/2$-subset of $[l]$. 

We start with an useful lemma that shows that the set of subspaces $S_1, ..., S_k$ do not have large intersections.

**Lemma 8** Let $C_1, ..., C_k$ be a set of matrices of order $l$, that satisfy the subspace property with the subspaces $S_1, ..., S_k$, then for any subset of indices $J \subseteq [k]$ \[ \dim(\cap_{i \in J} S_i) \leq \frac{l}{2|J|}. \]

Moreover, The number of subspaces $\{S_i\}_{i=1}^k$ that contain an arbitrary vector $v \neq 0$ is at most $\log l$.

**Proof:** Proof is omitted.

The previous Lemma shows that an arbitrary vector $v \neq 0$ can not belong to “too many” subspaces $S_i$. This observation gives a tight bound on the number of nodes $k$ in an optimal access code, as the following theorem shows.

**Theorem 9** Let $C_1, ..., C_k$ be a set of invertible matrices of order $l$ that satisfy the subspace property with subspaces $S_1, ..., S_k$. If each subspace $S_i$ is spanned by an $1/2$-subset of the standard basis $e_1, ..., e_l$, then $k \leq 2 \log l$.

**Proof:** Prove by Lemma 8 and a counting argument.

**Corollary 10** Let $k$ be the largest integer such that there exists an optimal access $(k + 2, k, l)$ MDS code, then $k = 2 \log l$.

**Proof:** The lower bound is derived by the code constructed in [1], [19].

Note that [19] constructed also an optimal bandwidth code with $k = 3 \log l$. Therefore, in the general case where we do not require an optimal update code, there is a difference between optimal access and optimal bandwidth code. Namely, these two properties are not equivalent (see Table I).

VI. SUMMARY

In this paper, we considered optimal bandwidth (resp. access) MDS codes with two parities. Specifically we asked, given the capacity of each node $l$ what is the largest possible integer $k$ such that there exists an optimal bandwidth (resp. access) $(k + 2, k, l)$ MDS code. We used distinct combinatorial tools to derive 3 upper bounds on $k$. The first bound considers the general case of optimal bandwidth code. The last two bounds are tight, and they consider optimal access and optimal update codes. Moreover, we showed that in the general case, the properties of optimal bandwidth and optimal access are not equivalent. Although in certain codes such as codes with diagonal encoding matrices, they are indeed equivalent. It is an open problem what is the longest optimal bandwidth code with capacity $l$.

VII. ACKNOWLEDGEMENT

This work was partially supported by an NSF grant ECCS-0801795 and a BSF grant 2010075.

REFERENCES


